This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2849420, IEEE Access

A Joint Multi-Feature and Scale-Adaptive Correlation Filter Tracker

XIAOHUI YANG[1], HAO ZHANG[1], LEI YANG[1],CHUNSHENG YANG[1,2] ,AND PETER X. LIU .[1,3] (Senior Member, IEEE)

[1]College of Information Engineering, Nanchang University, Nanchang 330031, Jiangxi, China
[2]National Research Council Canada, Ottawa, Ontario Canada
[3]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6

Corresponding author: xpliu@sce.carleton.ca

This work was supported in part by the National Natural Science Foundation of China (51765042, 61463031, 61662044), Jiangxi Provincial Department of Science and Technology (JXYJG-2017-02), Jiangxi Natural Science Foundation (20171ACB20007) and Jiangxi Provincial Department of Science and Technology (20121BBE50023, 20133BCB22002).

ABSTRACT

Traditional tracking-by-detection trackers may fail to track targets owing to interferences such as deformation, occlusion, fast or irregular motion, and background clutter. To improve the process of feature extraction, sample training, and the performance of the traditional kernelized correlation filter (KCF), this paper proposes a new joint multi-feature and scale-adaptive correlation filter tracking algorithm based on histogram of oriented gradients (HOG) and color-naming features. The new algorithm is composed of a position correlation filter tracker and a scale correlation filter tracker. It first uses a mask matrix to cut the training samples, obtains a higher proportion of real samples, and then trains the two correlation filter trackers. One is a position tracker that uses HOG and color-naming features to train two correlation filters, and then combines the results of the two correlation filters to calculate the target position. The second tracker is used to identify the scale with the maximum response size to select the target scale. The proposed algorithm achieves a competitive performance compared to algorithms such as KCF, compressive tracking (CT), and circulant structure kernel (CSK). The results show that this algorithm achieves a better performance when faced with occlusion, deformation, background clutter, and fast or irregular motion, and is able to track changes in target scale.

INDEX TERMS feature fusion, scale-adaptive correlation filters, target tracking, mask matrix

I. INTRODUCTION

RESEARCH into visual tracking algorithms based on tracking-by-detection [1]–[3] , is the focus of much activity in computer science. Calculating the position of a moving target in each frame of a given video sequence is a crucial aim of computer vision. With rapid developments in science and technology, target tracking technology has been applied in many fields and achieved good results [4]. A variety of high-performance tracking and detection algorithms are emerging, and their performance continues to improve. In real life, however, owing to many interferences such as target deformation, occlusion, fast or irregular motion, background clutter, and so on, precise positioning of the target is still very difficult to achieve.

Current target tracking algorithms can be divided into two kinds: the generating method and the discriminant method [5]. The generating method firstly constructs a model of the target, then searches the interesting image region for the region that is most similar to the model, and takes this region as the target region. However, in practice, the generating method has many defects and is unsatisfactory. In contrast, the discriminant method introduces machine learning into target tracking, and treats a target tracking task as a binary classification problem. With the introduction of machine learning, the discriminant method has been developing rapidly as the most favored method. Many excellent tracking algorithms now use discriminant methods, such as the Hedged deep tracking (HDT) algorithm [6], MOSSE filter [7], kernelized correlation filter (KCF) [8], CT algorithm [9], and Staple algorithm [10]. The correlation filter method was first introduced into the tracking realm with the use of the MOSSE filter to create a faster tracker. Some parameters for the MOSSE filter are computed in the Fourier domain. The authors exploited a fast Fourier transform (FFT) to improve tracker speed, used gray features to express the target, and used minimized quadratic variance to train the correlation filter. Finally, the tracker finds the position of the maximum response value in the response graph and takes it as a goal position. However, the MOSSE filter can only select a limited number of samples to train the classifier, which affects the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2849420, IEEE Access

performance of this correlation filter. Subsequently, Henriques proposed the circulant structure kernel (CSK) algorithm [11] and then developed the kernelized correlation filter (KCF) algorithm.Based on the MOSSE filter, the KCF algorithm uses a cyclic matrix and kernel trick to solve the problem that the filter can only select a limited number of samples. This method uses more samples to train the tracker, and improves the performance of the MOSSE filter in terms of accuracy and robustness. In the multi-channel, Henriques used histogram of oriented gradients (HOG) features instead of gray features, and achieved very good results. Although the algorithm has been applied in real life, there are still many problems [12]: (1) the algorithm has generally little effect in conditions of high-speed or irregular target motion; (2) the algorithm is less robust when faced with unexpected appearance changes due to occlusion or other interference; and (3) there is no mechanism to update the target scale.

In order to improve the performance of the KCF algorithm, we fuse HOG [13] features and color-naming [14] features to solve problems (1) and (2). Inspired by the literature [15], we employ the 31 oriented bins variant for HOG features. In the meantime, we use a mask matrix to cut the number of training samples and improve the accuracy of the real samples, and also to improve the algorithm's tracking accuracy. In order to solve the problem that the KCF algorithm has no mechanism to update the target scale, HOG features are used to train a scale correlation filter to detect changes in target scale. At the same time, an improved model update plan for the tracking algorithm is also proposed.

The main contributions of this paper are as follows. Firstly, we change the structure of the KCF algorithm. The proposed multi-feature fusion and scale-adaptive filter algorithm includes two correlation filters; one is used to track the target, and the second is used to estimate changes in target scale. Secondly, as features are a core component of the target, we propose a new feature-extraction method that fuses HOG features and color-naming features instead of a method based only on HOG. Finally, experiments show that the multi-feature fusion and scale-adaptive filter algorithm achieves an impressive performance in terms of both accuracy and robustness, and a better performance than most state-of-the-art tracking algorithms.

II. KERNELIZED CORRELATION FILTER

The kernelized correlation filter algorithm introduced the cyclic matrix and kernel method into the training of the regularized least squares (RLS) classifier to obtain a correlation filter, and successfully used the property of circulant matrices that all circulant matrices are diagonal in the Fourier domain. Converting matrix operations into operations of Hadamard product vectors greatly reduced the computational complexity and improved the computational speed compared to other tracking algorithms.

A. CYCLIC MATRIX

Assuming an interesting image patch, we use a one-dimensional vector to represent it, and then perform a cyclic shift operation to obtain all of the possible combinations:

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{1}$$

Then, all of the possible combinations are as follows:

$$\{P_X^u | u = 0, \cdots, n-1\} \tag{2}$$

There is a corresponding circulant matrix:

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_2 & x_3 & \cdots & x_n & x_1 \end{bmatrix} \tag{3}$$

As we know from Eq.3, the circulant matrix can only be determined by the initial sample. For all samples, all of the cyclic matrices are diagonal after the discrete Fourier transform (DFT). The circulant matrix can be expressed as

$$X = F diag(\hat{x}) F^H \tag{4}$$

In Eq. 4, F is the constant matrix, F^H is the Hermitian matrix for , and the symbol Λ represents the DFT.

B. RIDGE REGRESSION

Most tracking algorithms include three steps: training, detection, and update. For the training, we need to solve the ridge regression problem. A ridge regression function is always used in tracking algorithms since it has a simple closed-form solution [16] and good results can be obtained. Firstly, we analyze the linear regression. The goal of training is to find an objective function $f(z) = w^T z$. The linear regression problem can be formulated as follows:

$$min_w \sum_i (f(x_i) - y_i)^2 + \alpha \|w\|^2 \tag{5}$$

In Eq. 5, α is a regularization parameter that forms the penalty item with the norm, aiming to prevent over-fitting, and is the return goal. The closed solution of Eq.5 is given in the literature [16] as

$$w = \left(X^T X + \alpha I\right)^{-1} X^H y \tag{6}$$

In Eq. 6, T represents transposition, each row of matrix X is a training sample, and the first row is a reference sample; that is, the original sample x that has not undergone cyclic shift. Substituting Eq. 4 into Eq. 6 results in

$$w = F diag\left(\frac{\hat{x}^*}{\hat{x} \bigodot \hat{x}^* + \alpha}\right) F^H y \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2849420, IEEE Access

The symbol $*$ represents the conjugate, and $\odot$ indicates the dot product in the frequency domain,

$$Fz = \hat{z} \tag{8}$$

Substituting Eq. 8 into Eq. 7 affords the filter template as follows:

$$\hat{w} = \frac{\hat{x} \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \alpha} \tag{9}$$

C. KERNEL FUNCTION AND NONLINEAR REGRESSION

For the linear regression problem, we can use the above method to train the template directly, but it cannot deal with nonlinear regression. If we give sample z a linear transform, $f(z)$ is a linear function of the transformation results:

$$f(z) = w^T \phi(z) \tag{10}$$

From reference theorem, we can see that w is a linear combination of nonlinear $\phi(z)$, which can be expressed as

$$w = \sum_i \varphi_i \phi(z_i) \tag{11}$$

Combining Eq. 10 and Eq. 11 gives

$$f(z) = \sum_i [\varphi_i \phi(z_i) \phi(z)] \tag{12}$$

The kernel function is defined as $k(x_i, y_i) = \phi(x_i) \phi(y_i)$. Submitting the function into Eq. 12 gives

$$f(z) = \sum_i [\varphi_i k(z, z_i)] = \varphi^T k(z) \tag{13}$$

φ_i is a vector of $n \times 1$, $k(z)$ is a vector of $n \times 1$, and the element i is a kernel function value of a training sample and a test sample. As can be seen from Eq. 13, the nonlinear function $f(z)$ about z is a linear function about $k(z)$ after being mapped and introduced into the kernel function. The closed-form solution of the ridge regression problem can be solved using the method in Eq. 6 and the solution is

$$\varphi = (K + \alpha I)^{-1} y \tag{14}$$

In Eq. 14, K is the kernel matrix, and φ is the dual space solution of w . For the Gaussian kernel, we know from the relevant theorem that K is a cyclic matrix , so Eq. 14 can be written as follows, where $\hat{k}^{xx}$ is the first row of the kernel matrix K :

$$\hat{\varphi} = \frac{\hat{y}}{\hat{k}^{xx} + \alpha} \tag{15}$$

D. RAPID DETECTION

After the filter template is calculated by the previous steps, the candidate region is sampled. The candidate image patch is represented as z . Its size is the target size multiplied by 2.5, and the same cyclic shift is performed on z to obtain more candidate test samples. Using Eq. 13, the output response of all of the candidate test samples can be obtained as follows:

$$f(z) = (K^z)^T \varphi \tag{16}$$

In Eq. 16, K^z is the cyclic matrix corresponding to the kernel correlation between the training sample x and the candidate test sample z.

In the above equation, $f(z)$ is a one-dimensional vector containing the responses of all the candidate test samples, and the candidate image patch with the maximum response output is taken as the target position, which completes detection of the target.

III. MULTI-FEATURE FUSION AND SCALE-ADAPTIVE CORRELATION FILTERS

A. FEATURE EXTRACTION AND FUSION

HOG features have a high performance for target tracking when an image sequence exhibits variation in illumination and shadows, but their performance is less satisfactory when faced with fast or irregular target motion. The color-related features are not sensitive to deformation and have no boundary effect, but when illumination and background change they show an unsatisfactory performance for target tracking. Possegger et al. [17] proposed a distributed average tracking (DAT) algorithm that uses color histogram features, and literature [18] made the combination of HOG features and color features to achieve good results, showing that color features and HOG features can complement each other. In this paper, HOG features and the color-naming features of the target are extracted and used for the position filter training; that is, the position filter training is divided into two independent ridge regression problems. Firstly, HOG features are extracted from the target image and used to train a traditional kernelized correlation filter. The image block detected in the detection phase obtains the position with the highest response value given by Eq. 13. The color-naming feature was proposed by Danelljan et al. [14]. It consists of eleven different color channels and has a correspondence with the human visual senses. The validity of color-naming features has been verified in relevant documents. For the color-naming features in this paper, firstly, the training sample information in RGB color space is converted into color description features in 11-dimensional space using the method described by Weijer et al. [19]. The color-naming features are then extracted and used to train the correlation filter. In the detection phase, the tracker extracts the color-naming features of candidate samples and then obtains the location p with the largest response value as calculated by Eq. 13. At the same time, the sizes of the response values f_h and f_c are used to process the weights of the positions and , respectively. Some literatures

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2849420, IEEE Access

use the same weighting for positions p_h and p_c with the highest response, but many cases have shown that when HOG features are compared with color-related features, HOG features have more credibility and robustness for target tracking; hence, giving p_h and p_c the same weighting is less rigorous. In this paper we deal with the weighting as follows:

$$p = \partial_h p_h + \partial_c p_c \tag{17}$$

In Eq. 17, $\partial_h > \partial_c$ and $\partial_h + \partial_c = 1$, and p is the final response location.

The position correlation filter based on HOG features uses cyclic shift operations on real samples to construct more virtual samples [20]. However, the result of the cyclic shift is not a real sample [21], but a sample approximating a real sample. When encountering fast motion during occlusion and other issues, these virtual samples become unreliable. To solve this problem, we use a mask matrix to crop the training samples as follows. Let $X = (x_1, x_2, \cdots, x_n)^T$ be the training sample, and M the mask matrix. The size of M is $n \times (n-2)$ and $C(X)$ is the circulant matrix. This gives the following result:

$$M = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{18}$$

$$C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_2 & x_3 & \cdots & x_n & x_1 \end{bmatrix} \tag{19}$$

We cut the cyclic matrix $C(X)$ as follows:

$$M \times C(X) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_{n-2} \\ x_n & x_1 & x_2 & \cdots & x_{n-3} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-4} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_2 & x_3 & \cdots & x_{n-2} & x_{n-1} \end{bmatrix} \tag{20}$$

In Eq. 19, we can see that, after the original sample $n-1$ is rotated $n-1$ times, only the first of the n samples generated is a real sample, while the others are all synthetic samples. If we give the cyclic matrix and the mask matrix $n \times (n-2)$ multiplication operations, we can see that the first row and the second row will be real samples. This means that by using the mask on the large-sized cyclic shift matrix, we can enhance the proportion of the real sample, making a better filter template training.

B. SCALE IMPROVEMENT

The size of the kernelized correlation filter tracking frame is fixed and it is not possible to track changes in the target's scale. In literature [14], the target scale estimation and position estimation are fused into a correlation filter. Although this achieves an estimation of the target scale, a higher computational cost is required for this fusion. This paper is inspired by the discriminative scale space tracking (DSST) algorithm [14], which trains a scale tracker on the target.

The training for scale correlation filters is similar to the training of kernelized correlation filters. Taking the current target frame as a reference sample, HOG features are extracted from the reference sample and used as a training sample. The filter template is obtained from Eq. 15 to complete the scale tracker. The position tracker trained as above finds the new candidate position, and then uses a one-dimensional scale correlation filter to obtain the non-standard candidate image from the current center position. The extracted features are HOG features. The scale for this paper is 9, and the scale options are as follows:

$$a^n H \times a^n R, n \in \{-\frac{s-1}{2}, \cdots, \frac{s-1}{2}\} \tag{21}$$

In Eq. 21,H and R are the width and height, respectively, of the previous frame image, a is the scale factor, $s = 9$ is the scale number, and the above scales are nonlinear. The response is calculated using Eq. 12 for the candidate image, and the scale with the largest response is selected as the current target scale. The scale model is then updated as follows:

$$x_t = (1-\theta)\, x_{t-1} + \theta x_t \tag{22}$$

$$\mu_t = (1-\theta)\, \mu_{t-1} + \theta \mu_t \tag{23}$$

In Eq. 22 and Eq. 23, θ is a learning factor, x_t is the eigenvector of the tracking target, and μ_t is the coefficient of the detection model.

C. MODEL UPDATE

The updated scheme for the kernelized correlation filter model is shown in Eq. 22 and Eq. 23. It can be seen from the formulae that this model updating scheme only uses the information of the current frame and the previous frame. When the target is deformed, blocked, or rotated, it leads easily to a drift in the tracking process [22]. The literature [14] provides a general update from the first frame to the current frame.

For frame image t, the model coefficient is:

$$\mu_t = \frac{\sum_{i=1}^{t} \hat{y}\hat{K}}{\sum_{i=1}^{t} \left(\alpha + \hat{K}\right) \hat{K}} \tag{24}$$

Therefore, the model can be updated as follows:

$$\mu_n^t = (1-\theta)\, \mu_n^{t-1} + \theta \hat{K} \hat{y} \tag{25}$$

$$\mu_d^t = (1-\theta)\, \mu_d^{t-1} + \theta \hat{K} \left(\hat{K} + \alpha\right) \tag{26}$$

$$x_t = (1-\theta)\, x_{t-1} + \theta x_t \tag{27}$$

In the above equations, K is a kernel matrix and $\hat{y}$ is the output response.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2849420, IEEE Access

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The evaluation method in Wu et al. [23] has been widely used to evaluate tracking algorithms since it was proposed in 2013. In this paper, we use 50 datasets in OTB-2013 to test our algorithm and compare it with some representative algorithms stored in OTB-2013 to evaluate performance. The test environment was an Intel Core 2 Duo CPU i5 with a 2.50 GHz processor and 4 GB of memory. Some of the comparison algorithms were compiled in MATLAB 2016a environment.

A. PERFORMANCE EVALUATION METHODS

In order to evaluate the algorithm's performance, this paper adopts the evaluation method of OTB-2013. In quantitative analysis, we use the precision plot and the success plot. In the tracking accuracy assessment, the standard used in the literature [23] is the center position error, which is the average Euclidean distance between the center of the tracking target and the manually calibrated exact point. The mathematical expression is as follows:

$$CLE = \sqrt{(x_i - x_{in})^2 + (y_i - y_{in})^2} \tag{28}$$

In Eq. 28, CLE is the center position error,(x_i, y_i) is the target position as tracked by the tracker, and (x_{in}, y_{in}) is the exact position of the human mark. In the success rate assessment, the evaluation criterion is the overlap rate of the tracking frames, and the overlap rates greater than a given threshold are calculated. In the success rate graph, the abscissa is the overlap rate threshold, uniformly distributed from 0 to 1, and the ordinate represents the percentage of successful frames at this overlap rate threshold. The area under the curve is used for the success rate graph and to rank the tracking algorithm.

B. PERFORMANCE COMPARISON

In this experiment, we evaluate the algorithm by the traditional one-pass evaluation (OPE) method and compare it with the KCF algorithm and other excellent algorithms that exist in the OTB-2013 benchmark, such as CSK [11], ASLA [24],CXT [25], TLD [26], VTS [27], and LOT [28]. The results are shown in Figure 1. We can see that our algorithm ranks first in terms of accuracy and the success rate under different thresholds, in comparison with the algorithms in the OTB-2013 benchmark and KCF. From the results, we can see that the proposed algorithm achieves a tracking accuracy of 0.756, which is 7.8% higher than the traditional KCF algorithm. Its success rate is 0.650, which is 8.3% higher than the KCF algorithm. Its accuracy and success rate are a great improvement compared with the traditional KCF algorithm and other state-of-the-art algorithms.

Figure 2 and Figure 3 show the accuracy and success rate graphs for different tracking algorithms under different conditions. There are 11 kinds of target attributes in OTB-2013 that we test in this experiment. We list six important attributes in this paper as follows: background clutter, motion blur, occlusion, deformation, scale variation, and illumination variation. The comparison results show that the proposed algorithm outperforms the KCF algorithm in all six conditions and the comparison results show that the proposed algorithm outperforms the KCF algorithm and the other state-of-the-art algorithms in all six conditions.

Since the traditional KCF algorithm has no mechanism to update the target scale, it cannot track changes in target scale. Figure 4 shows the tracking results for the proposed algorithm in several sequences. From the figure, it can be seen that the proposed tracking algorithm in this paper can estimate changes in target scale.

The OTB-2013 benchmark contains 50 video sequences. We have chosen 22 representative video sequences to compare the proposed tracking algorithm with the KCF algorithm in order to account for its performance. Analysis of the tracking results shows that the proposed algorithm is significantly better than the KCF algorithm under a variety of conditions: background clutter, noise (shaking video sequences), deformation, occlusion, fast and irregular motion, and so on. In terms of tracking speed, the proposed algorithm is slower than the KCF algorithm. However, the average tracking speed is 23.08, which can meet the requirements of real-time tracking (see Table 1).

TABLE 1. Test results of precision and tracking speed (FPS) using OTB-2013 benchmark

	Precision(20pt)		FPS(20pt)	
	Ours	KCF	Ours	KCF
Coke	0.938	0.838	11.86	46.33
Deer	0.887	0.817	16.81	23.20
Dudek	0.909	0.877	13.32	25.44
FaceOcc1	0.962	0.730	14.19	38.67
FleetFace	0.620	0.460	16.62	31.88
Football	0.905	0.796	26.76	70.98
Freeman1	0.405	0.393	59.91	165.54
Freeman4	0.576	0.530	88.22	316.21
Ironman	0.197	0.217	11.79	57.53
Jumping	0.391	0.339	44.19	115.13
Lemming	0.475	0.495	21.88	29.04
Matrix	0.390	0.170	12.01	78.55
MotorRolling	0.043	0.049	12.99	36.28
Shaking	0.934	0.025	14.29	37.09
Singer1	0.963	0.815	10.59	30.53
Skiing	0.071	0.074	17.16	122.06
Soccer	0.704	0.793	19.68	25.69
Sylvester	0.885	0.843	22.65	47.52
Tiger2	0.562	0.356	19.75	36.11
Walking2	0.432	0.440	17.34	34.30
Jogging.1	0.974	0.235	15.27	68.77
Jogging.2	0.993	0.163	20.39	29.29
Average	0.646	0.475	23.08	66.64

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2849420, IEEE Access

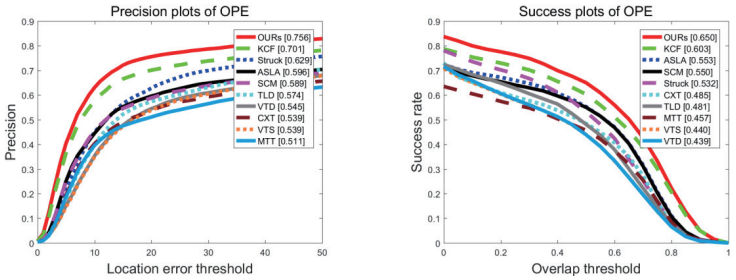

FIGURE 1. The accuracy algorithm map for different attributes.

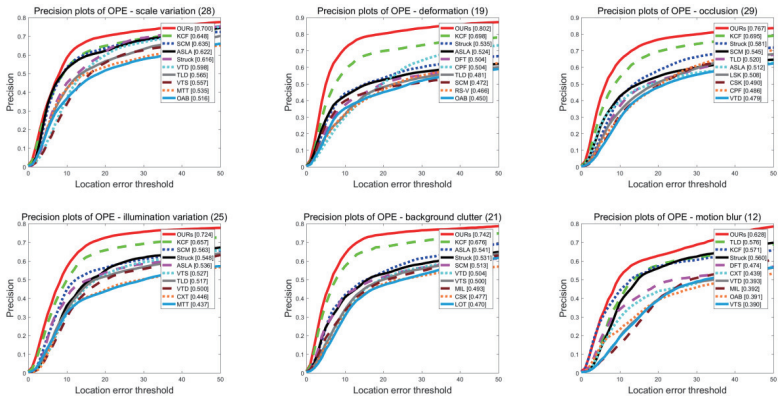

FIGURE 2. The accuracy algorithm map for different attributes

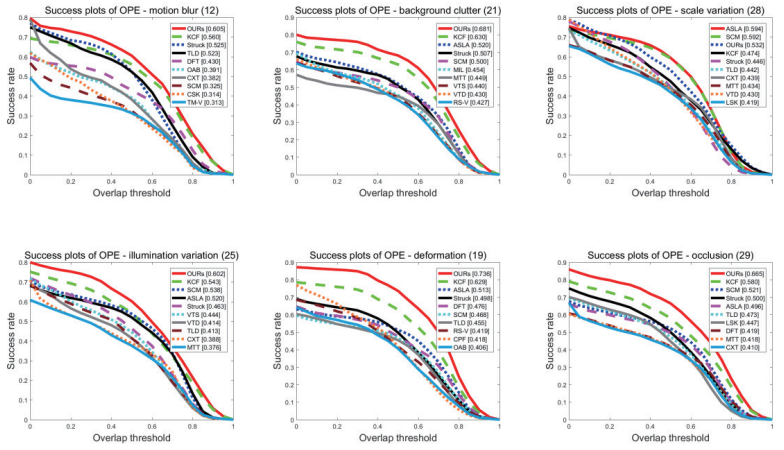

FIGURE 3. The algorithm success rate plot for different attributes

FIGURE 4. Tracking changes in the target scale

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have described some of the problems with state-of-the-art tracking algorithms, including the failure to track when certain conditions arise, such as target deformation, occlusion, irregular motion, jamming, and so on. These problems can be improved by a method that fuses HOG features and color-naming features. Since traditional tracking algorithms are unable to track changes in target scale, the proposed algorithm uses HOG features of the target to train a scale-adaptive correlation filter to estimate changes in target scale. At the same time, in order to improve tracker efficiency, we use a mask matrix when training trackers. By this method, the training samples are cropped, the proportion of real samples in the training samples is increased, and the performance of the tracking algorithm is improved. The experimental results show that the joint multi-feature and scale-adaptive correlation filter algorithm performs well under various challenging scenarios. A meaningful direction for further work is to find more robust features, and many more useful tracking algorithm structures may also be achieved from other state-of-the-art algorithms.

REFERENCES

[1] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 8, pp. 1619–1632, 2011.

[2] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 7, pp. 1409–1422, 2012.

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 2544–2550.

[4] X. Li, J. Zhang, and S. Wang, "The research on tracking algorithm of intelligent vehicle based on image recognition," in Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2017 14th International Computer Conference on. IEEE, 2017, pp. 101–105.

[5] Q. Liu, X. Zhao, and Z. Hou, "Survey of single-target visual tracking methods based on online learning," IET Computer Vision, vol. 8, no. 5, pp. 419–428, 2014.

[6] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4303–4311.

[7] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 2544–2550.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583–596, 2015.

[9] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in European conference on computer vision. Springer, 2012, pp. 864–877.

[10] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1401–1409.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in European conference on computer vision. Springer, 2012, pp. 702–715.

[12] F. Xu, H. Wang, Y. Song, and J. Liu, "A multi-scale kernel correlation filter tracker with feature integration and robust model updater," in Control And Decision Conference (CCDC), 2017 29th Chinese. IEEE, 2017, pp. 1934–1939.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 886–893.

[14] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press, 2014.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.

[16] R. Rifkin, G. Yeo, T. Poggio et al., "Regularized least-squares classification," Nato Science Series Sub Series III Computer and Systems Sciences, vol. 190, pp. 131–154, 2003.

[17] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2113–2120.

[18] J. Wang, W. Liu, W. Xing, and S. Zhang, "Visual object tracking with multi-scale superpixels and color-feature guided kernelized correlation filters," Signal Processing: Image Communication, vol. 63, pp. 44–62, 2018.

[19] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," IEEE Transactions on Image Processing, vol. 18, no. 7, pp. 1512–1523, 2009.

[20] H. Zhang and G. Liu, "Coupled-layer based visual tracking via adaptive kernelized correlation filters," The Visual Computer, vol. 34, no. 1, pp. 41–54, 2018.

[21] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in European conference on computer vision. Springer, 2016, pp. 419–433.

[22] E. Song, H. Lee, J. Choi, and S. Lee, "Ahd: Thermal image-based adaptive hand detection for enhanced tracking system," IEEE Access, vol. 6, pp. 12 156–12 166, 2018.

[23] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on. Ieee, 2013, pp. 2411–2418.

[24] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 1822–1829.

[25] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 1177–1184.

[26] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 49–56.

[27] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 1195–1202.

[28] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," International Journal of Computer Vision, vol. 111, no. 2, pp. 213–228, 2015.

XIAOHUI YANG received the B.Sc. M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China, in 2003, 2006, and 2015, respectively (email: yangxiaohui@ncu.edu.cn). He has been with the Department of Electronic Information Engineering, School of Information Engineering, Nanchang University, since 2006, where he is currently an Associate Professor. He has published over 30 research articles. His current interests include intelligent control, process control, fault diagnosis and stochastic nonlinear systems etc.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2849420, IEEE Access

HAO ZHANG received the B.E. degree in the measurement and control technology and instrument from East China University of Technology, Nanchang, China, in 2015, Now, he is pursuing the M.E. degree in electric engineering in Nanchang University, His research interests include machine learning and pattern recognition.

LEI YANG received the B.E. degree in Electrical engineering and its automation from Xi'an Polytechnic University , Xi'an, China, in 2016. Now, he is pursuing the M.E. degree in Electrical engineering in Nan Chang University.He research interests include New energy and Microgrid.

CHUNSHENG YANG is interested in data mining, machine learning, prognostic health management (PHM), reasoning technologies such as case-based reasoning, rule-based reasoning and hybrid reasoning, multi-agent systems, and distributed computing. He received an Hons. B.Sc. in Electronic Engineering from Harbin Engineering University, China, an M.Sc. in computer science from Shanghai Jiao Tong University, China, and a Ph.D. from National Hiroshima University, Japan. He worked with Fujitsu Inc., Japan, as a Senior Engineer and engaged on the development of ATM Network Management Systems. He was an Assistant Professor at Shanghai Jiao Tong University from 1986 to 1990 working on Hypercube Distributed Computer Systems. He was a Program Co-Chair for the 17th International Conference on Industry and Engineering Applications of Artificial Intelligence and Expert Systems. Dr. Yang is a guest editor for the International Journal of Applied Intelligence. He has served Program Committees for many conferences and institutions, and has been a reviewer for many conferences, journals, and organizations, including Applied Intelligence, NSERC, IEEE Trans., ACM KDD, PAKDD, AAMAS.

PETER X. LIU is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, K1S 5B6 Canada (email: xpliu@sce.carleton.ca). He received his B.Sc. and M.Sc. degrees from Northern Jiaotong University, China in 1992 and 1995, respectively, and Ph.D. degree from the University of Alberta, Canada in 2002. He has been with the Department of Systems and Computer Engineering, Carleton University, Canada since July 2002 and he is currently a Professor and Canada Research Chair. He is also with the School of Information Engineering, Nanchang University as an Adjunct Professor. His interest includes interactive networked systems and teleoperation, haptics, micro-manipulation, robotics, intelligent systems, context-aware intelligent networks, and their applications to biomedical engineering. Dr. Liu has published more than 280 research articles. He serves as an Associate Editor for several journals including IEEE Transactions on Cybernetics, IEEE/ASME Transactions on Mechatronics, IEEE Transactions on Automation Science and Engineering, and IEEE Access. He received a 2007 Carleton Research Achievement Award, a 2006 Province of Ontario Early Researcher Award, a 2006 Carty Research Fellowship, the Best Conference Paper Award of the 2006 IEEE International Conference on Mechatronics and Automation, and a 2003 Province of Ontario Distinguished Researcher Award. Dr. Liu is a licensed member of the Professional Engineers of Ontario (P.Eng), a senior member of IEEE and Fellow of Engineering Institute of Canada (FEIC).